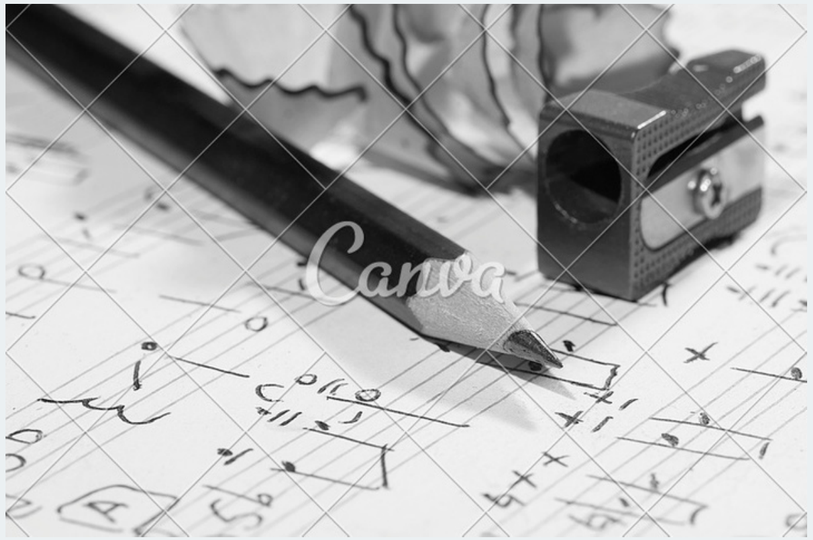
C♭

YVONNE CHEN | ISABELLA CHO | KATIE KIM | JASMINE VALERA

PROGRAMMING LANGUAGES AND TRANSLATORS

SPRING 2021

Language Overview




```
note n;
note v;
tone t;
octave o;
rhythm r;

n = ( /C-/ /4/ /s./ );

t = n.tone();
o = n.octave();
r = n.rhythm();

v.tone(t);
printt(v.tone());

v.tone(/A/);
printt(v.tone());

v.octave(o);
printo(v.octave());

v.octave(/5/);
printo(v.octave());

v.rhythm(r);
printr(v.rhythm());

v.rhythm(/h/);
printr(v.rhythm());

return 0;
```

Language Evolution

Iteration 0

- Vision: Easy-to-use composing tool for musicians who have limited programming experiences

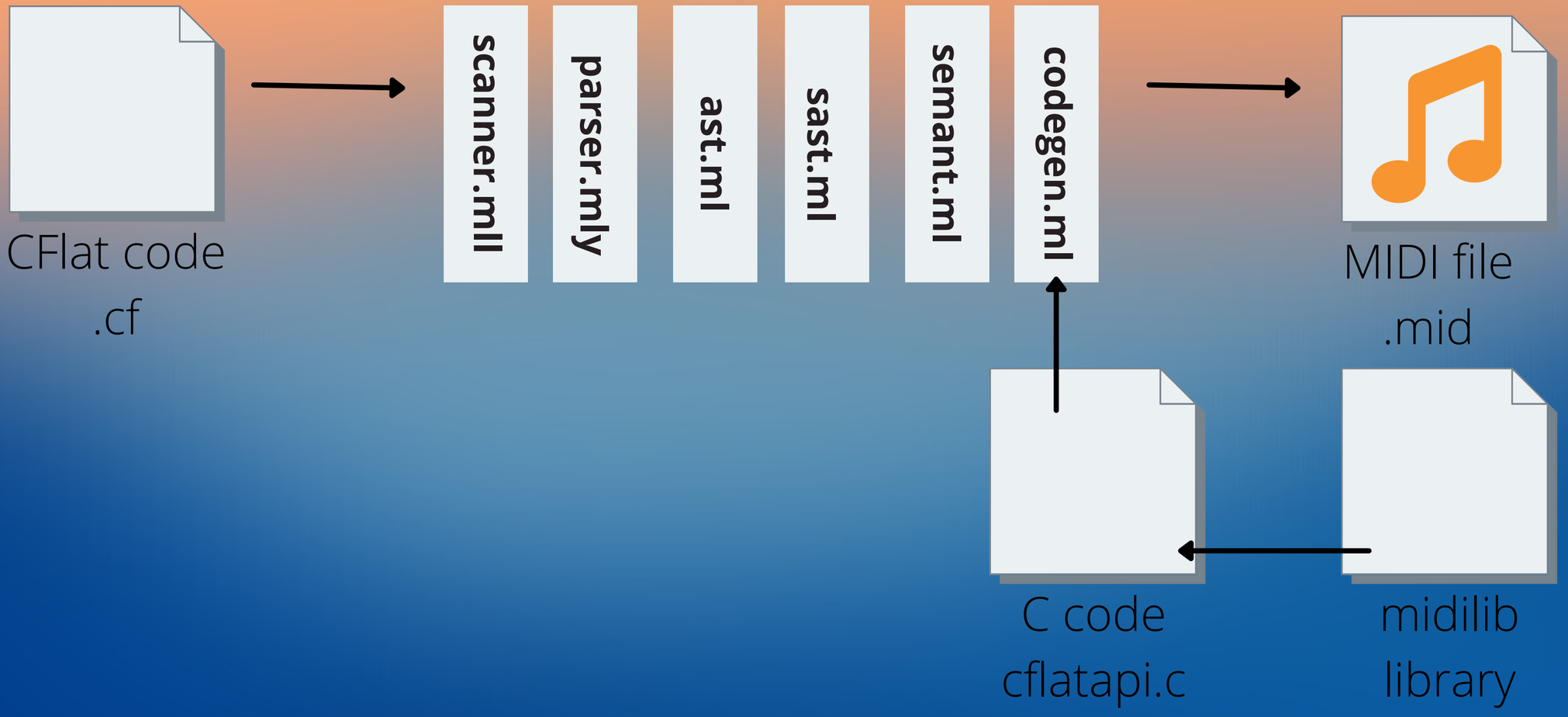
Iteration 1

- C-style syntax and control statements
- Note type that holds unique attributes (tone, octave, rhythm)
- API between CFlat and 3rd party MIDI library

Current Iteration

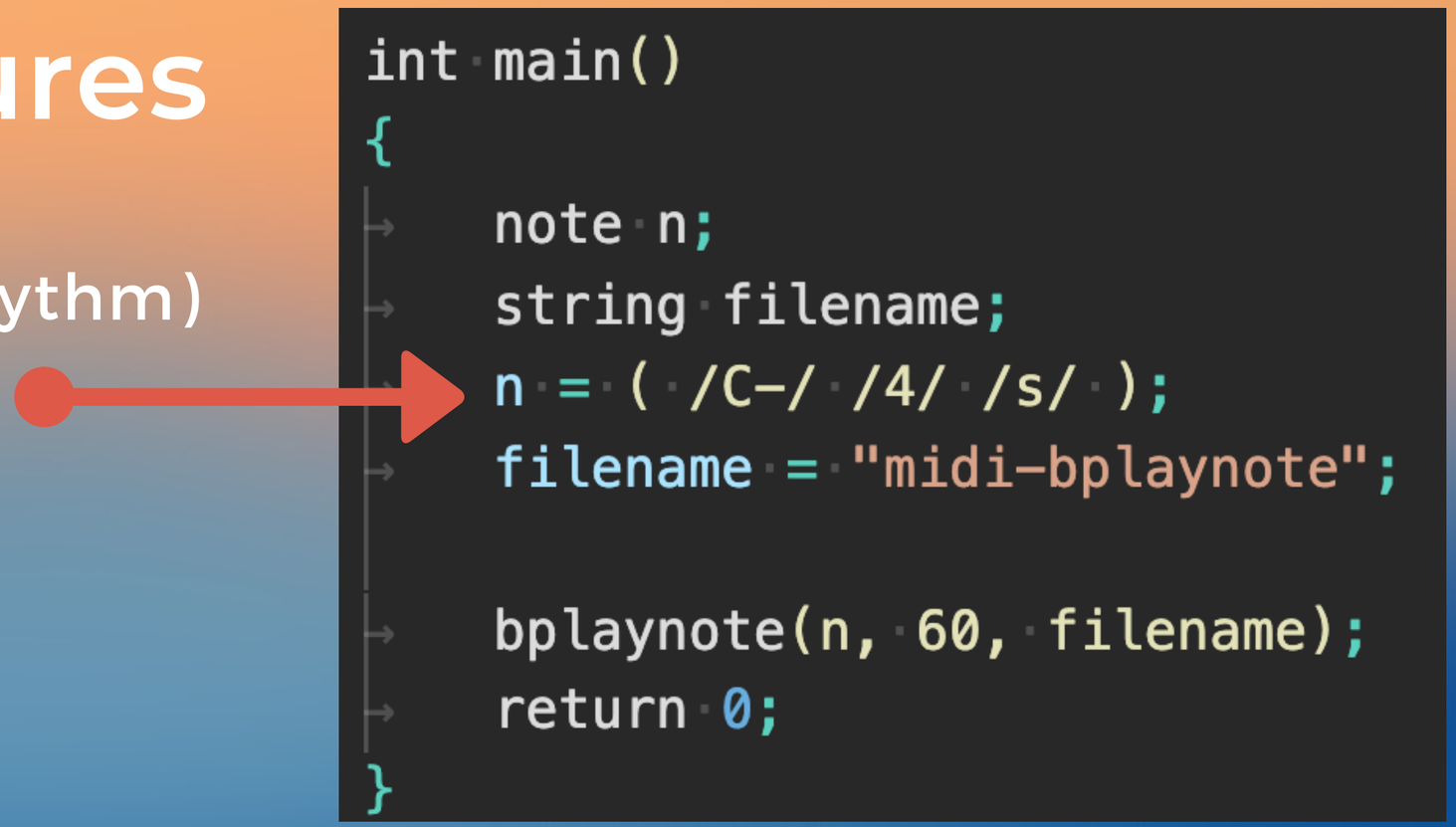
- playnote() and bplaynote() functions
- OOP approach to Note type (getter/setter-like functions)

Compiler Architecture



C ♭ Key Features

- note(tone, octave, rhythm)
- playnote()
- bplaynote()



```
int main()
{
    note n;
    string filename;
    n = ( /C-/ /4/ /s/ );
    filename = "midi-bplaynote";

    bplaynote(n, 60, filename);
    return 0;
}
```

Tone

- Note: struct
  - Tone: char*
  - Octave: int
  - Rhythm: char*

  - Tone: char*
    - /A/ /A-/ /A+/
    - /B/ /B-/ /B+/
    - /C/ /C-/ /C+/
    - /D/ /D-/ /D+/
    - /E/ /E-/ /E+/
    - /F/ /F-/ /F+/
    - /G/ /G-/ /G+/

Octave

- Note: struct
  - Tone: char*
  - Octave: int
  - Rhythm: char*

- Octave: int
  - /-1/
  - /0/
  - /1/
  - /2/
  - /3/
  - /4/
  - /5/
  - /6/
  - /7/
  - /8/
  - /9/

Rhythm

- Note: struct
  - Tone: char*
  - Octave: int
  - Rhythm: char*

  - Rhythm: char*
    - /s/ /s./
    - /e/ /e./
    - /q/ /q./
    - /h/ /h./
    - /w/ /w./

Rhythm								
Number of Beats	0.25	0.5	0.75	1.0	1.5	2.0	3.0	4.0
Letter Syntax	s	e	e.	q	q.	h	h.	w

Note Attribute Methods (GET and SET)

n.tone()
n.octave()
n.rhythm()

- takes in NOTHING-
- returns attribute -

n.tone(/A/)
n.octave(/4/)
n.rhythm(/h/)

- takes in ATTRIBUTE LITERAL-
- returns NOTHING -

Note Attribute Methods for OCTAVE

n.raiseOctave(i)

- takes in INT -
- returns NOTE with raised octave -
- return type: NOTE-

n.lowerOctave(i)

- takes in INT -
- returns NOTE with lower octave -
- return type: NOTE-

Demo: Note type

playnote(note n)

Create a playable MIDI file from a note type object

bplaynote(note n, int bpm)

playnote() with custom number of beats per minute

Demo: playnote()

Future Developments

- Array support for Note type & playarray()
- raise & lower methods for note attributes
- Harmony: MIDI with multiple tracks
- play() with custom instruments

References

On MIDI programming:

- https://github.com/MarquisdeGeek/midilib
- http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html#BMA1_3

Past projects:

- http://www.cs.columbia.edu/~sedwards/classes/2017/4115-fall/reports/Inception.pdf
- http://www.cs.columbia.edu/~sedwards/classes/2016/4115-fall/reports/Beathoven.pdf